复杂系统仿真模型在旅游纪念品市场分析中的应用

北京五中　高二七班　郭宇锋

目录

第 1 章　引言-----3
1.1　摘要-----3
1.2　问题的提出-----5
第 2 章　理论背景-----5
2.1　不完美信息静态博弈分析-----5
2.2　旅游纪念品市场的特征-----6
2.3　消费者的有限理性-----7
2.4　市场交易中的信息不对称性-----8
2.5　相关研究：伯川德模型-----9
2.6　遗传算法的改造-----10
第 3 章　研究目的及方法流程-----13
3.1　研究目的-----13
3.2　研究方法及流程-----13
第 4 章　不完美信息静态博弈模型-----14
4.1　模型背景-----14
4.2　模型的基本假设-----14
4.3　模型结果及分析-----14
第 5 章　基于 Java 语言的动态市场交易模型（UDTM）-----17
5.1　相关变量-----17
5.2　流程图-----18
5.3　流程-----18
5.4　程序的运行-----19
5.5　程序的结果-----20
5.5.1　原始机理模型-----20
5.5.2　完全信息完全理性模型-----22
5.5.3　完全信息完全理性模型（质量偏好）-----25
5.5.4　完全信息完全理性模型（价格偏好）-----28
5.5.5　非完美信息有限理性模型-----30
第 6 章　分析-----34
6.1　分析-----34
6.2　研究创新-----36
第 7 章　后记-----37
附录 A　原始模型 JAVA 代码-----38
附录 B　程序运行截图-----43
参考文献-----44

第 1 章　　引言

1.1 摘　要

在旅游纪念品交易市场中，经常会遇到“便宜没好货”的现象。为了回答为什么“便宜没好货”这一常识性问题，作者运用博弈论中的纳什均衡点概念对旅游纪念品市场进行了简单的静态博弈分析，随后运用 Java 语言编写了改进的遗传分类算法，进行了市场的复杂系统仿真模型的建立。在模拟了几种不同的消费者行为后，对结果进行了经济学的分析。

研究表明，市场信息的不对称性和消费者的有限理性是导致旅游纪念品市场出现“便宜没好货”现象的根本原因，且这两个因素广泛存在于市场中。这一现象的产生无法用传统的经济学理论进行解释，而在行为经济学的角度有一定的理论支持。在信息完全对称的情况下，且消费者绝对理性时，只有消费者和商品质量存在差异时市场才能出现均衡。在信息不对称的情况下，有限理性的消费者也可以达到均衡。且该假设符合实际情况。通过计算机建模的方式可以很好的观察市场的运行状态以及市场均衡形成的过程。

本研究的创新点及意义在于：

1、 运用静态博弈理论来研究旅游纪念品市场交易行为的均衡
2、 对这一日常生活现象从社会经济行为的角度进行思考
3、 运用 Java 编程构造市场系统，并融入了经过改造的遗传算法，以及独创的消费者函数
4、 将行为经济学的研究从心理学的范畴扩展到数学建模领域
5、 实现了传统数理经济学领域无法表示的消费者的有限理性和市场信息的不对称性的数学表达
6、 编程语言简单，可操作性极强。构造的模型不仅可用于研究旅游纪念品市场，因为所编写的程序是模块化程序，只需改变若干参数，替换消费者函数，便可作为其他市场状态下的机理模型，具有通用性

本研究中主要构造了如下 6 个模型：

1、 非完美信息静态博弈模型
2、 原始机理模型

3、 完全信息完全理性模型（价格，质量偏好）

4、 完全信息完全理性模型（质量偏好）

5、 完全信息完全理性模型（价格偏好）

6、 非完美信息有限理性模型

关键词：

旅游纪念品　复杂系统仿真模型　分类器算法　消费者行为

1.2 问题的提出

“便宜没好货”，从文字的角度来解释就是“价格低廉就没有质量好的东西”。从数学角度可将其定义为：价格低于市场均价，而质量低于一般商品平均水平的商品的销售行为。一直以来，消费者对于廉价商品的质量都存在着质疑[1]，也在每次的购买行为中进行着抉择。

在日常生活中，我也遇到过类似问题。比如，我经常在假期外出旅行，在旅游纪念品商店，会看到一些很有当地文化特色的商品。我都会选购一些感兴趣的旅游纪念品，送给亲朋好友。在进行购买时，作为普通消费者，我无从判断商品的质量，唯一能获取的信息就是靠价格。纪念品有高价和低价之分，每次我也都在两者中进行抉择。通常，我都会选择一些自认为价格合适的商品。回到家后，通过一段时间的观察，我发现购买的低价商品大多质量很差。这时，妈妈便会对我说：“便宜没好货”。通过与其他同学的交流，我发现这一生活现象普遍存在。

那么，在旅游纪念品交易市场中，为什么会出现“便宜没好货“的现象呢？哪些因素导致了这一现象的出现？它是否符合传统的市场经济学理论？能否从行为经济学的角度对其加以解释？又能否通过复杂系统仿真建模来观察这种现象并进行研究？

第 2 章　　理论背景

2.1　不完美信息静态博弈论分析

由于在旅游纪念品市场购买商品的过程中，商家与消费者在进行着利益的博弈。消费者希望以尽量低的价格买到尽量好的商品，而商家则希望获得尽量多的利润。双方围绕这一利益纷争点进行着博弈。因此从某种程度上来说，分析此问题可以从博弈论分析入手。

博弈论主要研究公式化了的激励结构间的相互作用，是研究具有斗争或竞争性质现象的数学理论和方法，也是运筹学的一个重要学科。 博弈论考虑游戏中的个体的预测行为和实际行为，并研究它们的优化策略。[3] 本文中所涉及到的不完美信息静态博弈，是博弈类型中的一种，其特点在于信息的不对称性。不完美信息博弈是指没有参与者获得其他参与者行动的信息。在本模型中指商家不知道消费者是否会购买，而消费者不知道商家提供的商品质量。

博弈论分析的方法多种多样，本文中所用的通过支付矩阵进行分析的方法堪称博弈论分析的经典方法。这一方法应用最经典的例子当属“囚徒困境”博弈。

囚徒困境的故事讲的是，两个嫌疑犯作案后被警察抓住，分别关在不同的屋子里接受审讯。警察知道两人有罪，但缺乏足够的证据。警察告诉每个人：如果两人都抵赖，各判刑一年；如果两人都坦白，各判八年；如果两人中一个坦白而另一个抵赖，坦白的放出去，抵赖的判十年。于是，每个囚徒都面临两种选择：坦白或抵赖。

从上述假设可以看出，此博弈中存在信息的不完美性，因为两囚犯被互相隔离，无法商量，所以不知道对方会采取怎样的行动。

运用博弈论的分析方法，首先列出支付矩阵

		囚徒 B	
		坦白	抵赖
囚徒 A	坦白	-8,-8	0，-10
	抵赖	-10,0	-1,-1

那么，不管同伙选择什么，每个囚徒的最优选择是坦白：如果同伙抵赖、自己坦白的话放出去，不坦白的话判一年，坦白比不坦白好；如果同伙坦白、自己坦白的话判八年，不坦白的话判十年，坦白还是比不坦白好。结果，两个嫌疑犯

都选择坦白，各判刑八年。如果两人都抵赖，各判一年。

从上述博弈分析中可以看出，此博弈存在一个稳定的均衡点，即双方都坦白。但是还存在另一个均衡点，即双方都抵赖，这样双方的获益总和是最大的。之所以没有达成这一均衡是因为信息的不完美性。倘若双方能够串供的话，恐怕均衡点就会出现在（-1，-1）上了。所以在考虑博弈问题时，一定要注意问题的背景给模型假设带来的影响。所以应先研究模型的背景特点，并依据此制定模型的假设。

2.2 旅游纪念品市场的特征

旅游纪念品，即是游客在旅游过程中购买的精巧便携、富有地域特色和民族特色的工艺礼品，并让人铭记于心的纪念品。

旅游纪念品市场交易的特征有如下几点[5]：

1、 当在其他地方进行消费时，如超市或菜市场，商品是消费者所熟悉的，也大概知道实际价值，因此可以根据这些信息做出比较合理的判断。然而在纪念品商店所销售产品对顾客来说是不熟悉的，并且商家一般会通过精美的包装和极具说服力的推销来让顾客进行购买，使顾客不能很好的判断商品的质量。因此具有很强的信息不对称性。

2、 与那些消费者定期就会去消费的场所相比，游客在旅游中几乎不会重复到一个地方，所以在一家纪念品商店购买的产品很难会进行退换，因此商品的质量对于商家的销量影响较低。

3、 消费者在日常消费中，购买商品的冲动性较轻。然而当人们在旅游时，购买纪念品的动机一般为纪念动机，是一种典型的情感型购物行为，一般注重的是通过消费获得精神的愉悦，舒适，个性的满足，所以消费者对于商品的价格反应也不很敏感。

从上述特征中不难看出，与其他消费场所如菜市场或是超市相比，旅游纪念品市场有其独特的一面。在这个市场中，商家在商品信息上占有巨大的优势，同时又通过其不靠回头客，以及消费者在对商品不了解的情况下所进行的冲动消费来获得高额利润。消费者在这个市场中由于上述原因，呈现出一种与传统经济学假设所相悖的“有限理性”行为。与此同时，信息不对称也是该市场突出的特征之一。

2.3 消费者的有限理性

从上述分析可以明显看出，厂商的生产及定价行为和消费者的购买行为与消费者的“有限理性”现象的出现密切相关。而传统的经济学模型都假设消费者的购买行为符合“理性人”的假设。

传统经济学理论赖以存在的基石便是“理性人”的假设。他，可以“像爱因斯坦一样思考，且具有相当于 IBM 电脑的记忆容量，以及同甘地一样强大的意志力”（36.5°行为经济学）。从上述描述不难看出，理性人具有以下特点：

1、 他具有极高的智慧，可以进行复杂的获益运算
2、 他记忆力超强，可以回忆起从前所有的交易行为情况
3、 他可以抛弃一切情感带给人的影响，一切评价标准都是极其客观量化的
4、 他唯一的目的就是在约束条件下获得最高的利益

伴随着这一根本思想，经济学的先驱亚当斯密提出了“我们做事情的原动机就是利己主义”这一经典主张。在当时的生产条件及人类认知的情况下，这一主张似乎很好的解释了人们生产及交易活动的本质。

有限理性的概念最初是阿罗提出的，他认为有限理性就是人的行为“即是有意识地理性的，但这种理性又是有限的”。一是环境是复杂的，在非个人交换形式中，人们面临的是一个复杂的、不确定的世界，而且交易越多，不确定性就越大，信息也就越不完全；二是人对环境的计算能力和认识能力是有限的，人不可能无所不知；此外，在很大程度上，由于受到情境的影响，人们使用“第一系统”进行加工，理性在这里根本就未发挥作用。[2]

并且，随着近十年来行为经济学逐步获得认可后，人们注意到，人的有限理性本身就是人与生俱来的一种属性。人类绝非是绝对理性的或者利己主义的动物。这一观点已得到广泛的认同。

因而，在本次建模中，对于“理性人”的存在对市场能否起到如理论中所阐述的影响，也是本课题研究的目的之一。本课题中我对于“理性人”的描述就是，不受自己所获得的其他信息的干扰，只专注于有关自己偏好的利益的信息，并只对这方面的信息做出反馈。

2.4 市场交易中的信息不对称性

在第 2 项的描述中，我阐述了信息不对称在旅游纪念品市场中存在的依据，

下面就来介绍一下信息的不对称性。

市场经济发展了几百年，都是处于不对称信息的情况之下，当人们没有发现信息不对称理论的时候，比如亚当 •斯密的时代，市场并没有显示出多么的缺陷，斯密甚至把“看不见的手”推崇备至，自由的市场经济理论学者都宣扬市场的自由调节，反对对市场的干预。

当信息经济学逐步深入人心后，信息的不对称性逐步得到大多数经济学家的认可。

但是，从现有的研究可以看出，信息经济学是基于对现有经济现象的实证分析得出的结论，对于解决现实中的问题还处于尝试性的研究之中。例如，买者对所购商品的信息的了解总是不如卖商品的人，因此，卖方总是可以凭信息优势获得商品价值以外的报酬。交易关系因为信息不对称而形成了博弈关系，交易中拥有信息优势的一方为卖方，不具信息优势的一方是买方。交易双方实际上是在进行无休止的信息博弈。

将信息的不对称性融入模型中，反映实际问题，便是研究的目的之一。

2.5 相关研究：伯川德模型[4]

在查阅了相关文献后，我找到了一个与本文研究相关的模型——伯川德模型。伯川德模型是由法国经济学家约瑟夫 • 伯川德于 1883 年建立的。

一、模型假设

伯特兰德模型假定，当企业制定其价格时，认为其他企业的价格不会因它的决策而改变，并且 n 个（为简化，取 n=2）寡头企业的产品是完全替代品。A、B 两个企业的价格分别为 P1、P2 ，边际成本都等于 C。

二、模型内容

在上述假设成立的情况下，设市场的需求函数为 Q=a-bp （a,b 为常数，p 为价格）由于两个厂商之间物品为完全替代品，所以价格高的厂商将失去整个市场，而价格低的厂商将得到整个市场，所以利润函数变为

$0<p1<p2$ 时：$Profit1=(p1-c)(a-bp1)$

$p1=p2$ 时：$Profit1=(p1-c)(a-bp1)$

$0<p2<p1$ 时：$Profit1=0$

三、伯川德悖论

根据上面推导的结果，可以得到著名的伯川德悖论：

1.寡头市场的均衡价格为：P=MC；

2.寡头的长期经济利润为 0。

这个结论表明只要市场中企业数目不小于 2 个，无论实际数目多大都会出现完全竞争的结果，这显然与实际经验不符，因此被称为伯川德悖论。

悖论的解法主要有两种，一是认为企业的生产能力有限，不可能供应整个市场的需求。二是认为当产品出现差异时，厂商产品之间的完全替代关系就消失了，同样可以解开悖论。

该模型假设的突出特点就是为了简化研究而使商家生产完全同质的商品并简单地界定了消费者对于不同市场状况的反应。该模型在消费者绝对理性的前提下模拟了一个完全竞争的市场模型。

然而，我个人认为，在这个模型中，经济学家明显没有考虑到另一个重要的因素：消费者。当引入有不同偏好的消费者时，同样可以解开伯川德悖论。并且，就算产品出现差异，倘若消费者之间不存在差异的话，他们仍然会做出同样的判断，这与之前悖论的内容就一致了。也就是说，产品和消费者都要存在差异才能解开悖论。这也是我此次研究要论证的问题之一。同时，该模型中消费者在给定的信息的前提下做出了完全理性的决定，然而在我的模型背景中消费者是存在“有限理性”的。也就是说，我希望通过某种途径将消费者的“有限理性”表现出来。

2.6　遗传算法的改造

伯川德模型虽然很好的构造出了一个市场交易模型，但其模型假设与我所要构造的模型假设差异很大。我需要构造出一个聪明的、追求利益的、会学习的商家；一个有着有限理性的消费者群体；一个信息不对称，商品存在质量差异的交易平台。

在上述思路的指导下，结合之前查阅的文献和做出的分析，我想到了通过计算机进行系统仿真模拟，并选择了 Java 语言作为编程的平台。那么，如何在模型中模拟出商机对于利润的追求呢？如何表现出商机通过尝试不同的决策从而积累经验，逐步优化自己的策略，最终与消费者达成某种均衡呢？

为了模拟公司的定价机制，我想了很多种算法。由于交易市场中存在很大的

随机性，所以厂商利润的最大化属于数学中非线性优化问题。所以我开始查阅非线性优化算法，最后我查阅到了有关遗传算法的材料。在深入理解遗传算法的基础上，我对它进行了有机的改造，去除掉了遗传算法中交叉，变异，淘汰，替换等算法内容，单单保留了分类器算法。

分类器算法属于高级人工智能[6]，它能使厂商具有追逐最大利益的能力，并能不断完善自己的策略，对于各种市场情况都有相对应的解决方法，这对应着每一个市场状态的解决方法就叫做一个分类器。

1、分类器的特征值赋值

为了辨别市场状况，每一个分类器都有一个特征值，特征值定义的规则如下：

分为 4 条规则 a：产品价格最近是上涨了还是下跌了 B：销售量是上涨了还是下跌了 c:最近利润上涨了还是下跌了 d:质量是优质还是劣质

若上升，赋值为 1，下降或不变，赋值为 0

这样，每一个分类器都拥有一个特征值，0000,0001……当一轮交易结束后，将会生成一个市场状态变量，之后根据这个变量与特征值进行比对，选择对应的分类器。

2、分类器如何做出决策

遗传分类算法为每一个状态赋予一个概率向量

P1:降价的概率

P2:涨价的概率

p3：价格不变的概率

p4:提供优品的概率

p5：提供劣品的概率

为防止遗传分类算法出现训练过度的情况，会给每种状态赋予最大值和最小值的限制。

进入到某种状态时，公司会根据相应的概率向量生成随机数来决定策略，在交易完成后再根据利润的变化来调整策略

例如，对于（1,0,0,1）(p1,p2,p3,p4,p5)=(0.1,0.3,0.6,0.5,0.5) 而生成了随机数，0.2,这时应当价格不变。然后再生成随机数，0.4，这时应提供优质商品。如果最终利润增加了，则策略会调整为（0.05,0.35,0.55,0.6,0.4）。使利润增加的决策发

生的概率增加了，厂商做出更明智的决策的能力得到了加强。

反过来，如果决策导致了利润持平或下降，则上轮做出的决策的概率将会下降，其他决策的概率将会上升。根据人类学习的过程，厂商通过试错，得到经验，学习，最终得到进化，变成更会赚钱的厂商。这符合实际中厂商学习如何做出决策的行为。

第 3 章　　研究目的及方法流程

3.1 研究目的

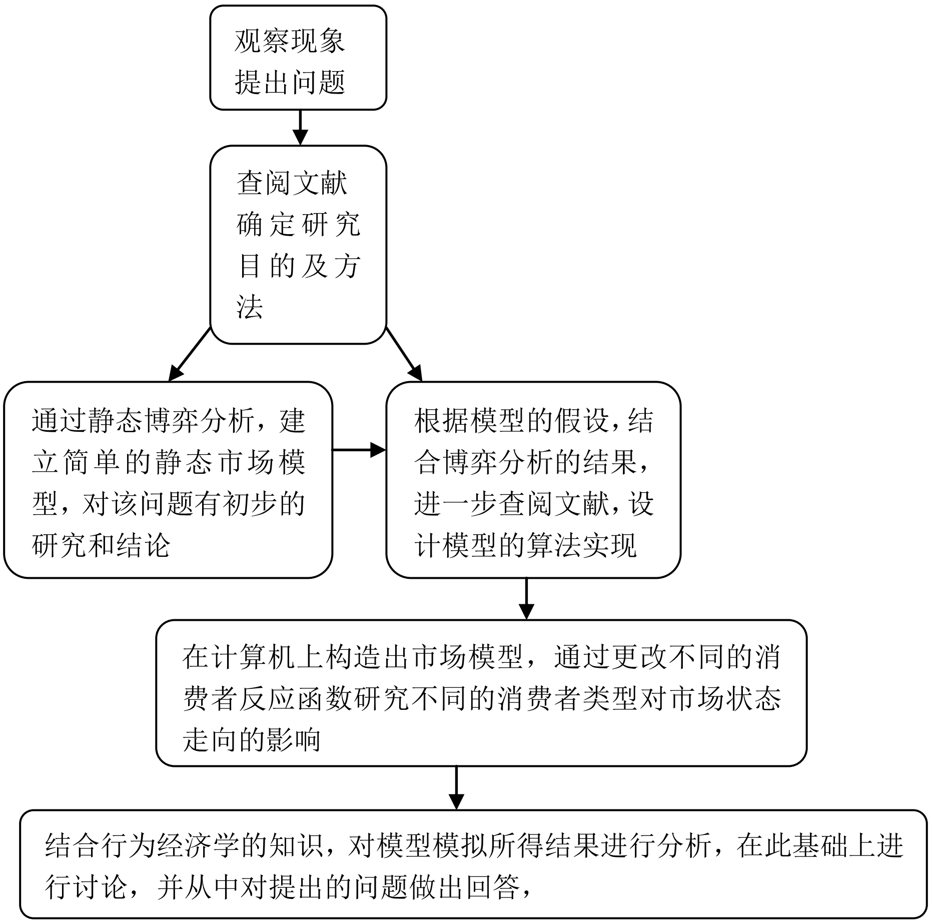
1、 探究导致旅游纪念品市场低价劣品泛滥的原因

2、 探究如何将有限理性和信息不对称融入数学模型中

3、 研究不同模型结果所对应的经济学结论

4、 构造出旅游纪念品市场的复杂系统仿真模型

3.2 研究方法及流程



第 4 章　　不完美信息静态博弈模型

4.1　模型背景

基于上述知识背景的了解，我建立了旅游纪念品市场交易的博弈模型。不完美信息博弈的本质特征在于博弈方之间在信息上是不对称的。以该模型为例，由于卖家拥有更多的信息，特别是卖家为了提高销量，一定会不遗余力的伪装产品的质量，使消费者不容易直接判断出产品质量的优劣，因此买方具有不完美的信息。这种关于产品质量信息上的不对称，使得买方在交易开始便处在一种不利的地位，也是低价劣品泛滥于旅游纪念品中的原因。

4.2　模型的基本假设

1、 只有一个厂商提供产品，每个消费者只买一次，每个阶段只有一个消费者，消费者的策略是决定是否购买商品，厂商的策略是决定是提供高质量还是低质量产品

2、 消费者在购买时不知道自己所买产品的质量，只知道产品的价格。厂商在出售商品时知道自己产品的质量

3、 假设市场上的产品只有劣品和正品两种，成本为 c1,c2**(c1<c2)**产品价格有高价和低价两种，售价为 p1,p2**(p1<p2)**

4、 消费者分为购买和不购买两种情况，消费者对于优质商品及劣质商品的评价分别为 g 和 b 当买方以 p1,p2 的价格购买正品时，获得的消费者效用分别为（g-p1,g-p2）,且**(g-p1)>0,(g-p2)>0**；当买方以价格 p1 和 p2 购买劣品时，获得的消费者效用为(b-p1)和(b-p2)，且**(b-p1)>0,(b-p2)<0**,且 **g>b**;不购买商品时，消费者效用为 **0**

注：消费者效用即消费者通过购买行为获得的收益，消费者效用**=**（消费者评价-价格）

4.3　模型结果及分析

1、 低价出售

		消费者	
		买	不买
厂商	劣品	P1-c1,b-p1	-c1,0
	优品	P1-c2,g-p1	-c2,0

在此博弈中，纳什均衡解为（低价，劣品，买）

分析：

依据上策最优法，可得出纳什均衡：

<u>厂商：以次充好，降低成本增加利润，制造劣品</u>

<u>消费者：根据价格判断是否购买，低价买入劣品</u>

从上表中可以看出，当厂商决定以低价出售商品时，消费者无从获得商品质量的信息，而厂商也不知道消费者是否会进行购买，符合信息的不对称性。双方同时开始进行博弈，在这个静态模型当中，（低价，劣品，买）是一个纳什均衡解。厂商获得的收益也很明显，厂商以低价销售时，会选择劣质商品。消费者选购劣质商品的原因是无法判断质量，仅从价格上判断一定会选择低价的商品，所以消费者选择购买低价的商品。

这个均衡极不易破坏。对于商家来说，消费者购买低价商品，他若从劣质品改为提供优质品势必会增加成本，降低利润，所以他不会改变策略。对于消费者来说，只能获得价格的信息，所以选择高价商品会增加自己付出的价格，减少消费者效用，所以消费者会选择低价商品。又因为市场中消费者和商家无法进行信息的交换，所以上述分析全部成立。在此情况下，任意一方单方面改变自己的策略都会导致自身获益的减少，所以双方都不会改变自己的策略。这一思想符合纳什均衡点的概念。

高价出售

		消费者	
		买	不买
厂商	劣品	P2-c1,b-p2	-c2,0
	优品	P2-c2,g-p2	-c2,0

在此博弈中，纳什均衡解为（高价，劣品，不买）

分析：

同上，依据上策最优法得到均衡。

<u>厂商：高价出售提供劣品，获得高利润</u>

消费者：高价劣品，不买

从上表中可以看出，在这个静态博弈中，（高价，劣品，不买）是一个纳什均衡解。对于厂商来说，即使高价出售，也愿意提供劣品，毕竟劣品的成本比优品成本低，可以获得更高的利润。而消费者选择不买，是因为若消费者以高价购买了劣品，则效用将为负。（高价，优品，买）相对于纳什均衡解来说效用更高，但联系到“囚徒困境”中不难发现，在信息不对称的前提下，由于对对方的不信任，这个更优的策略组合对彼此的约束力很小，因而这一均衡解几乎不可能达成。双方陷入“囚徒困境中”，是市场完全失败的均衡。不禁令我想起“柠檬原理”，劣币驱逐良币，劣品驱逐优品。

综合以上两个均衡进行比较，可以明显得出：相对于第二个均衡，第一个均衡所对应的厂商获益和消费者效用都更多。因此，在产品市场上，当商品价格较低时，消费者购买到劣品的可能性非常大。而当高价出售时，商家和消费者无法形成交易，导致卖方撤出市场，市场陷于瘫痪。这也是美国经济学家阿克罗夫在1970年提出的“柠檬原理”的现象。

第 5 章　　基于 Java 语言的动态市场交易模型（UDTM）

UDTM(universal dynamic transaction model)是通用动态市场交易模型的缩写。我运用 Java 语言成功写出了该模型的代码。

该模型的核心思想是仿真模拟一个旅游纪念品交易市场，该市场有两个主体：商家和消费者。两个主体处在一个市场的环境中，通过交易发生交互，并不断生成反映市场状态的变量：价格，销量，利润，质量。每一轮销售行为中，这四个变量都会输出。通过一定天数的交易后，观察价格变动的情况，分析价格的走势与消费者反应函数之间的关系。对于能形成在一定区间内波动的均衡价格的情况，将对其他变量进行更深入的研究。

该模型分成以下几部分：

变量声明

参数录入

分类器算法定价系统

消费者反应函数

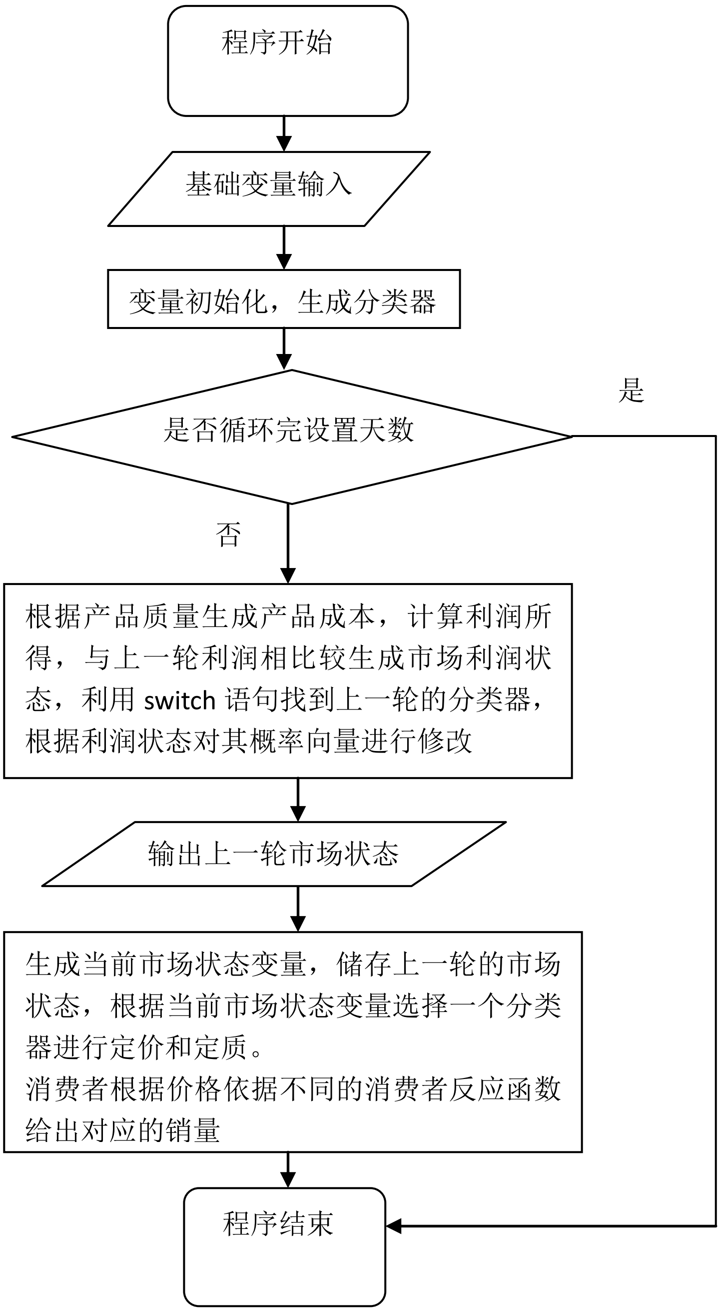
结果输出

下面将以最简单的原始机理模型进行算法运行的解释：

5.1　相关变量

为了维持本套算法的运行，声明了 18 个变量，所有变量及参数的值都可以修改。需要录入的基础变量有以下几个：消费者数量，原始价格，原始销量，原始质量，交易天数，优品成本，劣品成本

5.2 流程图



5.3 流程

1、 输入基础变量

2、 销量，质量，价格，市场状态变量初始化

3、 生成一个 16 乘 6 的二维数组，一个横行对应一个市场状态，竖行分别对应涨价，平价，降价，优品，劣品的概率及适应度 6 个数值，一个横行称为一个分类器

4、 开始进行市场行为模拟，根据天数设置循环次数

5、 先进行商品质量的判断，根据质量生成产品成本

6、 利润的计算，销量乘以售价减去成本

7、 根据与上一轮利润的比较来生成市场利润状态

8、 运用 switch 语句，找到上一轮决策的对应的分类器，并根据市场利润状态对 5 个概率向量进行修改。此处设置有一检验器，当某一概率向量大于最大值或小于最小值时，有一套算法对该分类器的概率向量进行修改，防止某一向量出现负值

9、 市场状态初始化

10、 计算市场状态变量的值，价格上涨则+8，销量上涨则+4，利润上涨则+2，质量降低则+1，若不符合都不进行计算，由此可得到 16 种市场状态，且每一种状态都对应 1 个唯一的市场状态变量值

11、 将销量，质量，价格，利润存入上一轮销量，质量，价格，利润变量中，方便下一次决策前的比较

12、 输出上一轮的 4 个变量

13、 根据市场状态变量的值找到 2 维数组中的分类器，先生成第 1 个随机数，进行首次判断，是否涨价，若涨价则再生成 1 个随机数，判断质量好坏。若不涨价，则再判断是否平价，若平价则再生成一个随机数判断质量。若不涨价也不平价，则直接生成随机数判断质量。每一种状态判断完后，都会生成 3 个量，当前价格，质量，以及决策变量。此处有一检验器，当价格小于成本或销量为负值时，有一套算法保证价格最小值等于成本，销量最小值等于 0

14、 销售环节，根据不同的销量反应函数给出销量。回到步骤 5，继续循环。

15、 循环结束后，输出 2 维数组，观察各情况的概率向量值以及适应度的值

5.4 程序的运行

程序编写应用的是 TextPad，运行系统为 windows xp 32-bit 由于结果输出界面有大小限制，在多次尝试后，我选择将循环次数定为 50 天，这样可以观测到均衡形成的过程。所有结果都与运行 20000 轮后的结果相同。

5.5 程序的结果

类型	反应函数	说明	均衡状态
机理模型	nSales=nCustomer	所有的消费者都购买，消费者无差异	达到均衡，价格波动小
信息完全对称，完全理性模型	if(nQuality==1){for(j=1;j<nCustomerA+1;j++){probability=Math.random(); if(probability<0.9){nSales+=1;} }}//质量偏好的消费者进行购买 if(nPrice<nPreviousPrice){for(j=1;j<nCustomerB+1;j++){probability= Math.random();if (probability<0.9){nSales+=1;} }}//价格偏好的消费者进行购买}	消费者存在不同的偏好，并根据自己所掌握的信息进行不同的购买行为，有差异	达成均衡，价格，销量，质量，利润波动大
完全偏好质量模型	if (nQuality==1){nSales=nPreviousSales+5;} else {nSales=nPreviousSales-5;}	消费者偏好质量，并获得了质量的信息，无差异	达成均衡，但利润为零
完全偏好价格模型	nSales=nCustomer-(nPrice-nOriginalPrice);	消费者偏好价格，并获得了价格的信息，无差异	未达成均衡
非完美信息有限理性模型	if (nPrice>nMaxPrice){nSales=nCustomer/2;} else {nSales=nCustomer;} if (nPrice<nMinPrice){nSales=nCustomer*2;}	消费者不了解质量，有一个价格区间，并对价格做出反应，无差异	达成均衡，且符合现实中的市场状态

下面对各模型及其结果做进一步详细的介绍。

5.5.1 原始机理模型

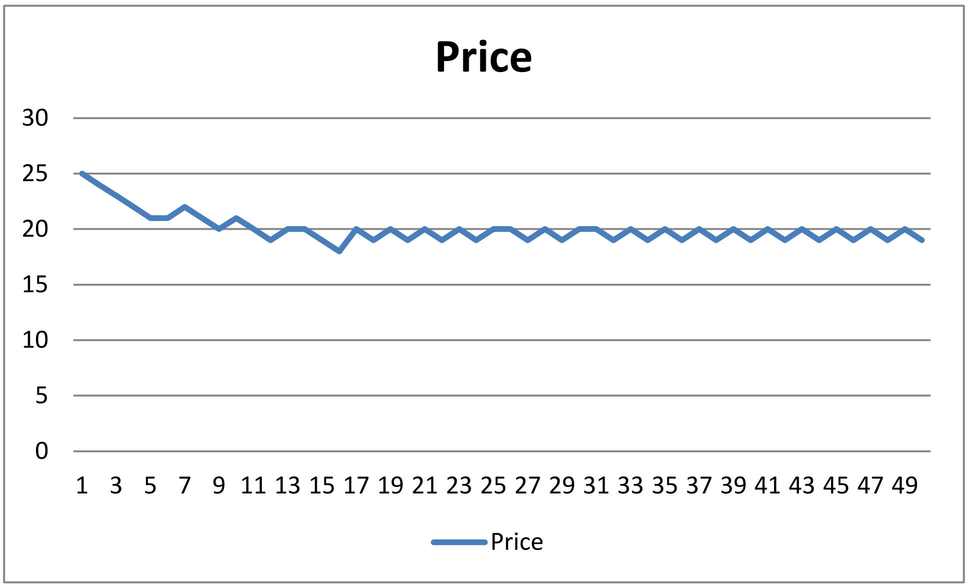
首先编写出来的是原始的机理模型，它的消费者函数很简单

nSales=nCustomer

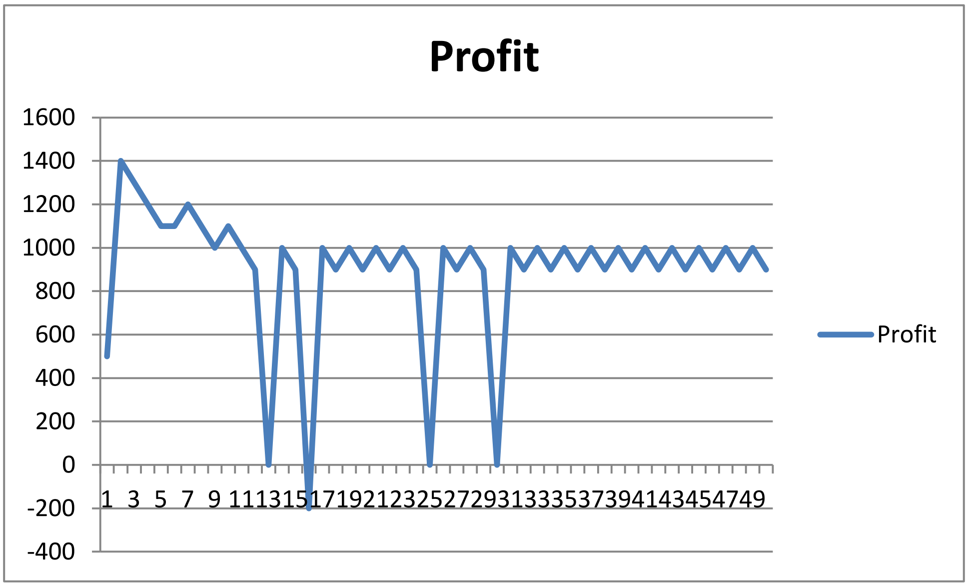
从上面这个数学表达式可以看出，这个函数反应的是需求弹性为 0 的消费者进行购买时的行为，即有多少消费者就有多少销量。这一特点体现了信息的不对称性：即销量表达式中没有价格，消费者不知道价格的信息。还体现了消费者的有限理性：价格的变动对于销量毫无影响，而“理性人”会对不同的价格作出不同的购买行为。

参数设置为：消费者：100，初始价格：25，初始质量：优质，初始销量：100，优品陈本：20，劣品成本为 10，价格变动幅度：1

该程序运行的结果如下：



从该折线图可以看出，价格初始有大幅下滑。但是在 17 天后，价格逐步趋于稳定，且稳定在 20 至 17 之间呈周期波动。可以看出，波动的周期在不断减小。因此可以得出这样的结论：分类器算法很好的起到了学习优质策略的作用，商家是向着正确的方向学习的，且其与该型消费者之间的博弈有均衡，且均衡价格位于 20 至 17 之间，远小于初始价格，介于优品和劣品成本之间，但远高于劣品。



从该幅折线图不难看出，初始成本从 1 天到 16 天之间有大的波动，且交易过程中有 1 次出现了负的利润，而这在厂商追求最大利益的前提下是不现实的，所以在后面的模型中我加入了检验器算法，即当价格小于成本时自动将价格调整为当前的成本。

单单看价格和利润图，暂时无法分析出负利润出现的原因。不过可以很清晰的看出，17 天之后利润出现稳定，与价格开始稳定的天数相同。之后虽然还有 2

次利润为 0，但除去这 2 次，利润已经稳定在（900,1000）的区间中。进一步说明了我改进的遗传分类算法的学习能力是正确并有效的。

参考本张折线图，很容易解开上面遗留的疑问，即负利润出现的原因。负利润出现在 16 天，恰好这一天商家提供了优质的商品，因此必然是在这一天价格低于成本。而其他 3 次利润为 0 都是因为出现了优质的商品。可以看到，在 31 天后，再也没有出现过优质商品。生产优质商品的策略被逐步弱化，生产劣质商品的策略在不断加强。

在需求弹性为 0 的条件下，市场的状态出现了（低价，劣品，买）的均衡，这种情况符合前面静态博弈模型的结果以及柠檬原理的推论。结果一方面说明了系统的学习能力。另一方面，该模型的一大特点就在于它的反应函数模拟了信息完全不对称，消费者有限理性的情况。由此可以得出，信息完全不对称，消费者有限理性的情况下，会出现（低价，劣品，买）的均衡。然而从另一个角度分析，在真实的市场条件下，消费者无法得到价格的信息这一现象是根本不可能存在的。所以该模型所作出的结论并不符合真实的市场状况。

5.5.2　完全信息完全理性模型

与第一个模型相对应，第二个模型所强调的是信息的完全对称以及消费者的完全理性。

消费者反映函数是通过 Java 语言中的判断语句实现的：

```
if(nQuality==1){for(j=1;j<nCustomerA+1;j++){probability=Math.random();
if(probability<0.9){nSales+=1;} }}//质量偏好的消费者进行购买
```

if(nPrice<nPreviousPrice){for(j=1;j<nCustomerB+1;j++){probability= Math.random();if (probability<0.9){nSales+=1;} }}//价格偏好的消费者进行购买}

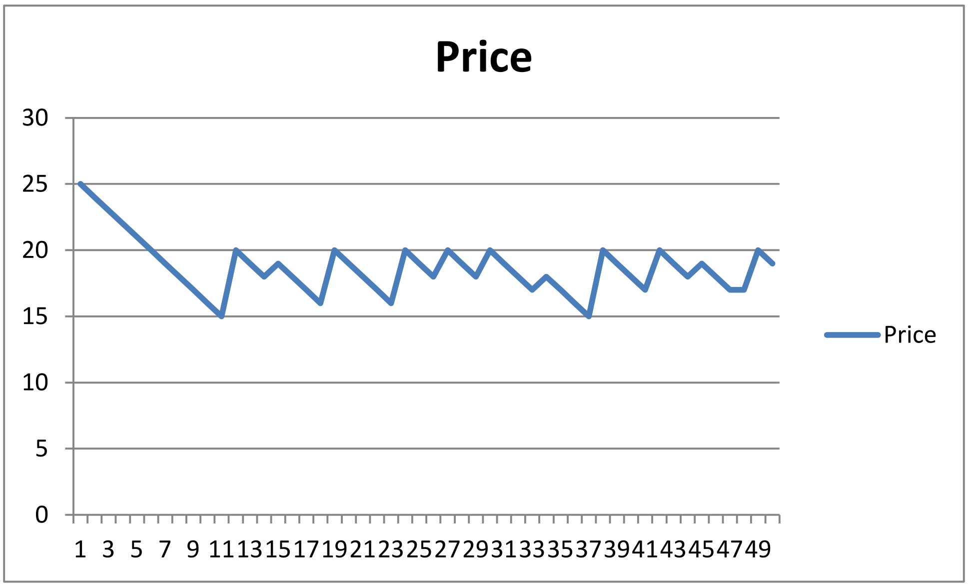
该函数定义了两种消费者，偏好质量的消费者和偏好价格的消费者。偏好质量的消费者，他所追求的理性的目标就是买到质量最优的商品。他虽然得到了市场关于价格的信息，但是他并不追求性价比，他只看重质量。所以，他只有在优质商品出售的时候才会购买。与之相对应的，偏好价格的消费者所追求的是更低的价格，他虽然得到了质量的信息，但为了达到更低价格的目的，他选择了不对质量信息做出反馈的策略。因此，价格偏好的消费者只有在价格比上一轮低的前提下才会进行购买。

本函数模拟的意义在于，我认为“理性人”所追求的物质利益最大化，可以在不同类型的消费者身上得到具体且有差别的体现。对于质量偏好的消费者来说，这个就是质量的最大化。对于价格偏好的消费者来说，这个就是价格的最低。

为了模拟真实，加入了随机情况，但是只要随机数小于 0.9 消费者就进行购买，也可以体现出只要符合自己物质上的利益就进行购买的“理性人”的假设。

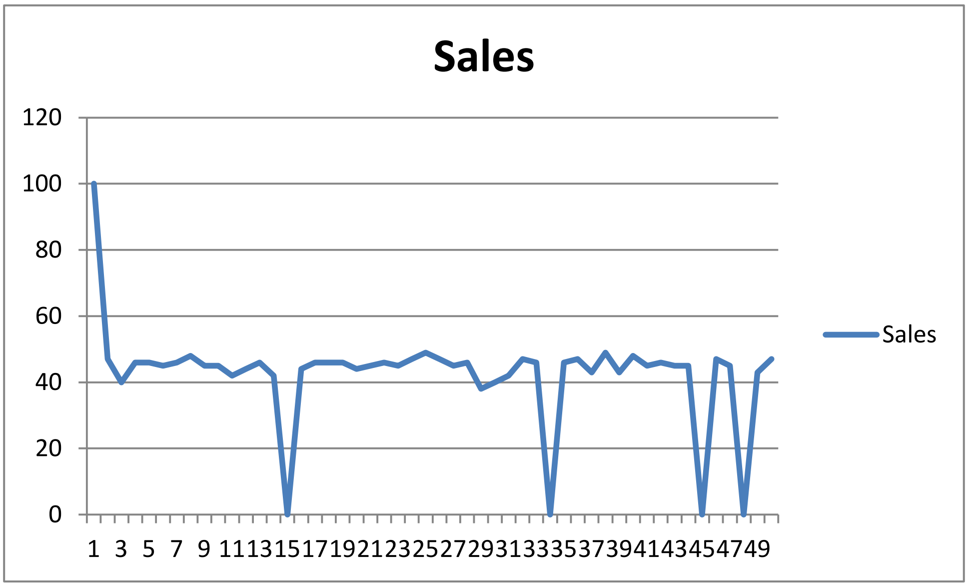
参数设置为：偏好质量的消费者：50，偏好价格的消费者：50，初始价格：25，初始质量：优质，初始销量：100，优品陈本：20，劣品成本为 10，价格变动幅度：1

程序运行的结果如下：

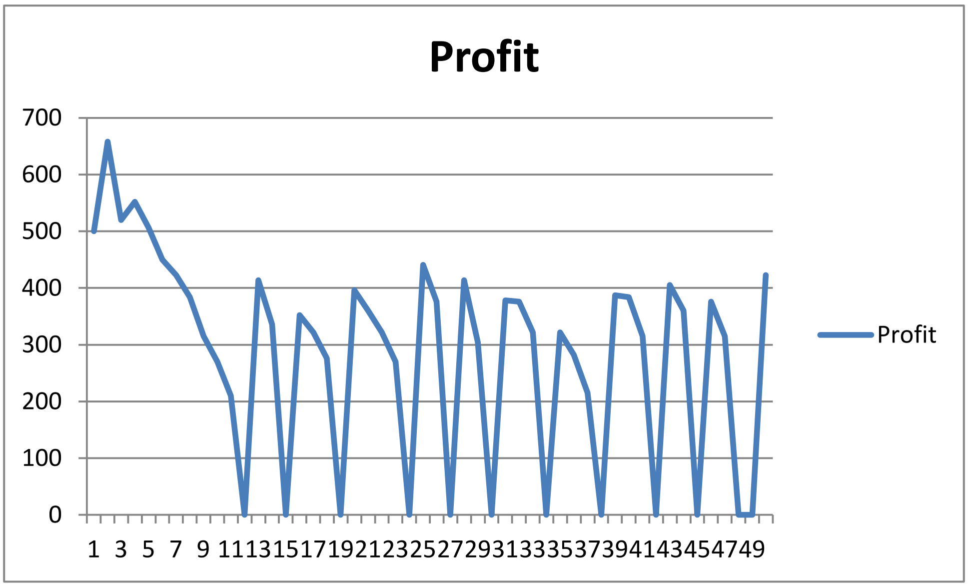


从折线图中可以看到，市场价格先大幅下降，之后便在（15,20）区间内波动，目前无法看出周期是否在缩小。总的来说，平均价格在 17 左右，低于优质

商品的成本，但高于劣质商品的成本。通过与第一个模型的价格折线图进行比对，可以看出，当市场中存在等量的不同偏好的完全理性消费者时，市场价格波动的周期和幅度都更大，不容易形成均衡。但价格的上涨和下降存在周期，是在有规律的波动。也可以理解为，形成了动态的均衡。

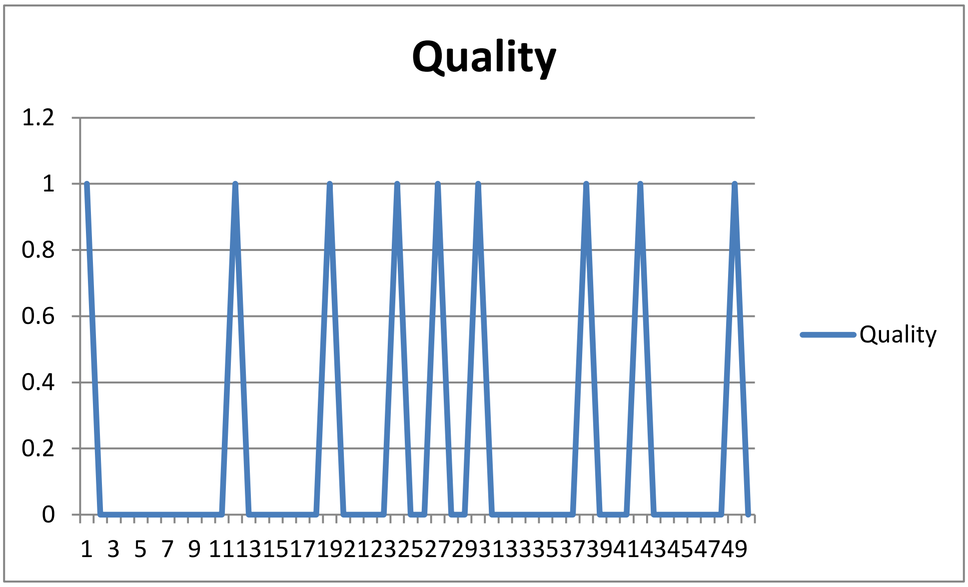


通过观察销量可以看出，首先前 3 天销量迅速下降，然后保持均衡。平均销量在 40 到 45 之间。除了 4 次有销量为 0 的情况外，销量总体水平保持稳定，可以说在销量上市场形成了稳定的局面。



从本图可以看出，利润的情况不容乐观，且波动较大。可以看到，在第 9 天后，利润出现周期性的上涨下降，且最大利润都在 400 左右，最小利润在 0 附近。且在 40 天的时间内出现了 11 个周期，可见利润变动的周期之短。这与前面价格和销量的周期性波动密切相关。现在看来，虽然没能形成一个稳定状态的市场，

但本模型模拟出来的更像是一个商业周期。随着需求量的周期性上升，下降，利润也周期性的上升，下降。由此可见，本模型的消费者反应函数符合古典经济学“商业周期”模型中理性消费者的描述。



从图中可以看到，商品的质量也和价格，销量，利润一样出现了周期性的波动。通过与利润的比较，每当出现优质商品时，利润都会下降到零。从图中可以看出，质量变动的周期有增长的趋势。因此可以预测，利润变动的周期也可能延长，最终市场可能会形成一种周期均衡，即古典经济学对“商业周期”的描述。

初步猜测该现象的出现是因为存在有不同种类的消费者，每当厂商做出一种决策时，都会有对应的消费者做出购买行为。当质量偏好的消费者进行购买时，厂商利润都为零，所以下一轮厂商不会采用这种措施。又因为市场中存在这样的顾客，所以为了满足需求，只能周期性地满足两种顾客。然而，这种周期在这个模型中出现的内在原因是什么，还有待于进一步的研究。

5.5.3 完全信息完全理性模型（质量偏好）

为了回答研究之初所提出的在市场中引入不同偏好的“理性人”是否会对市场最终的均衡结果造成影响的问题，我在第 2 个模型思路的基础上构造了第 3 个模型：消费者拥有完美的信息，具有完全的理性且偏好质量的消费者。

本模型的消费者反应函数如下：

```
if (nQuality==1){nSales=nPreviousSales+5;}
else {nSales=nPreviousSales-5;}
```

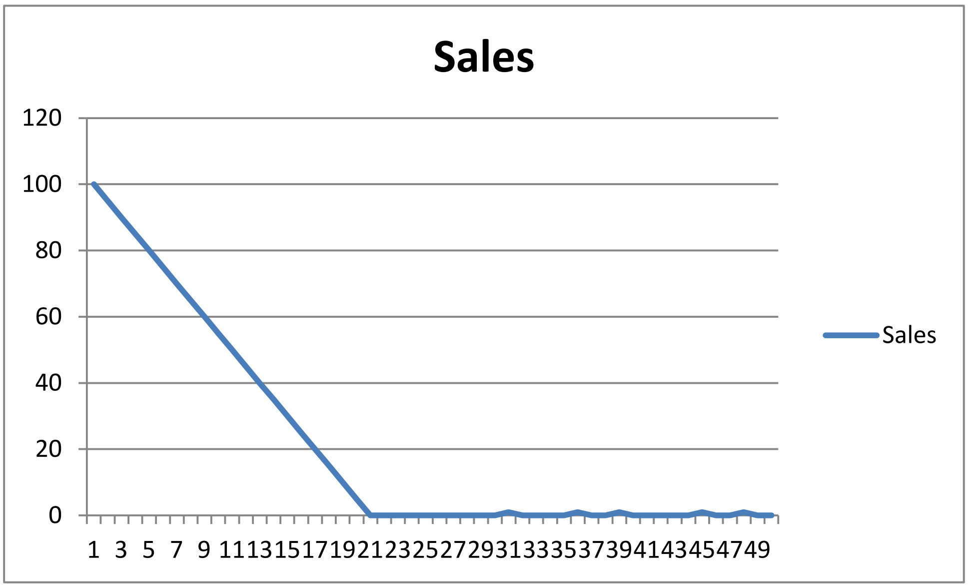
如果本期的质量为优质商品，则本期销量将比上期多出 5 个单位；若为劣

品，则销量将较上期减少 5 个单位。在这个函数中，每一个消费者都得到了关于价格的信息，并且在商家推出优品的时候会理性地增加购买量，体现了“理性人”的思想。

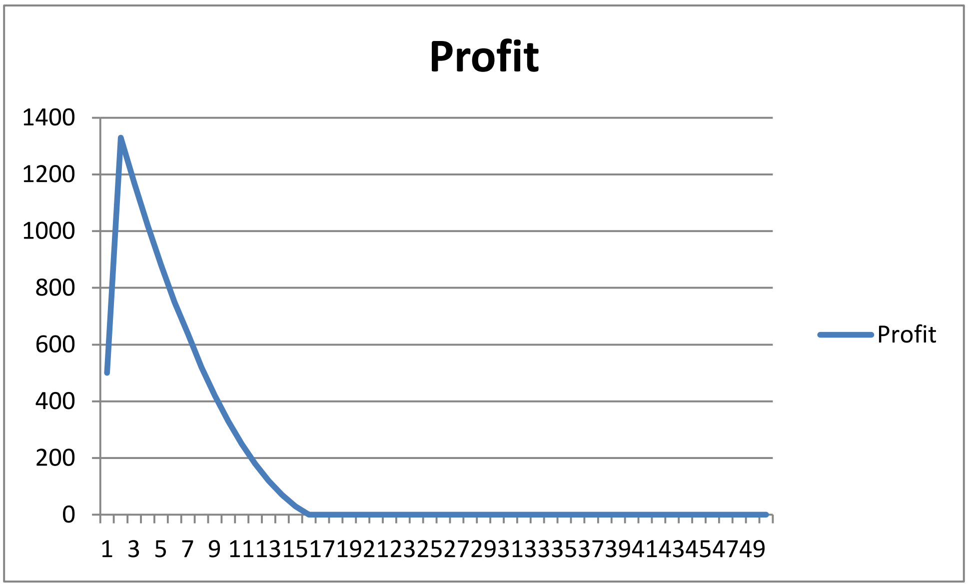
参数设置为：消费者：100，初始价格：25，初始质量：优质，初始销量：100，优品陈本：20，劣品成本为 10，价格变动幅度：1

程序运行结果如下：

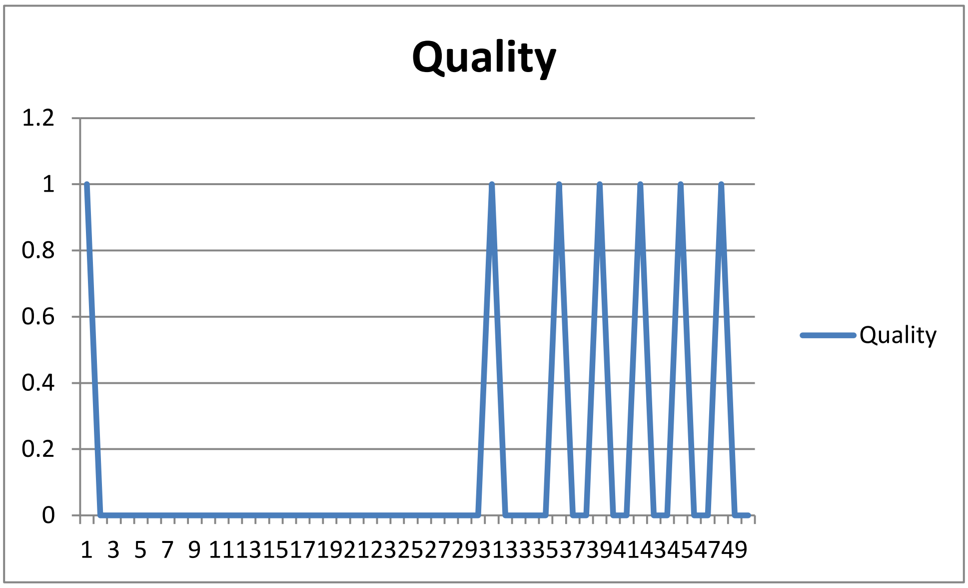
这个价格曲线的趋势比较明显，价格从第 1 天一直下降到第 15 天，直到劣品的成本，随后又保持这一价格，直到 29 天又上涨，随后在 20 到 17 之间波动，形成一个比较稳定的均衡价格。我猜测，初期价格的下降很有可能是质量偏好的消费者由于低质量的商品而拒绝购买，商家只能通过降价来获得市场份额。然而唯一能让消费者实施购买行为的途径就是提高价格，因此我猜测在 29 天后商家开始提供优质的商品。同时为了保持利润，又没有一直提供优质的商品，而是劣品和优品周期性提供，因此导致了价格的周期性波动。



通过观察销量折线图，可以印证通过价格统计图分析出来的一些结论。首先，在第 1 天到第 15 天，销量是一直下降的。然而，在 21 天后销量竟然有一段时间为零，之后的销量也只是比零高一点，从第 21 天后几乎没有售出商品。



从图中可以明显看出，除了第 2 天利润有大幅上升外，从第 2 天到第 15 天利润直线下降，然后一直为零。这一图像与销量图像比较吻合。



从质量图中可以看到，从价格图中分析得出的结论基本得到了证实，在 29 天后，商家开始周期性提供优质商品以挽救市场。然而，从市场给出的销量图可以看出，市场并不买账。只有在提供优质商品时才有少量的购买，而且利润一直为零。从该模型可以看出，本模型是出现优质商品次数最多的模型，可以看出，“理性人”的偏好决定了市场的供给。然而，在这样一个信息对称，消费者理性的市场中，商家的利润竟然为零，销售额也几乎为零。可以说，这个市场是一个失败的市场。这也印证了行为经济学的一些观点，“理性人”存在于市场中本身是不合理的。信息的不对称性和消费者的有限理性才是市场机制发挥其作用的根本所在。

5.5.4 完全信息完全理性模型（价格偏好）

与第 3 个模型相对应，第 4 个模型模拟的是市场中全部为偏好价格的消费者时市场的情况。本模型中，所有的消费者都知道价格的信息，且完全的理性，偏好价格。反应函数如下：

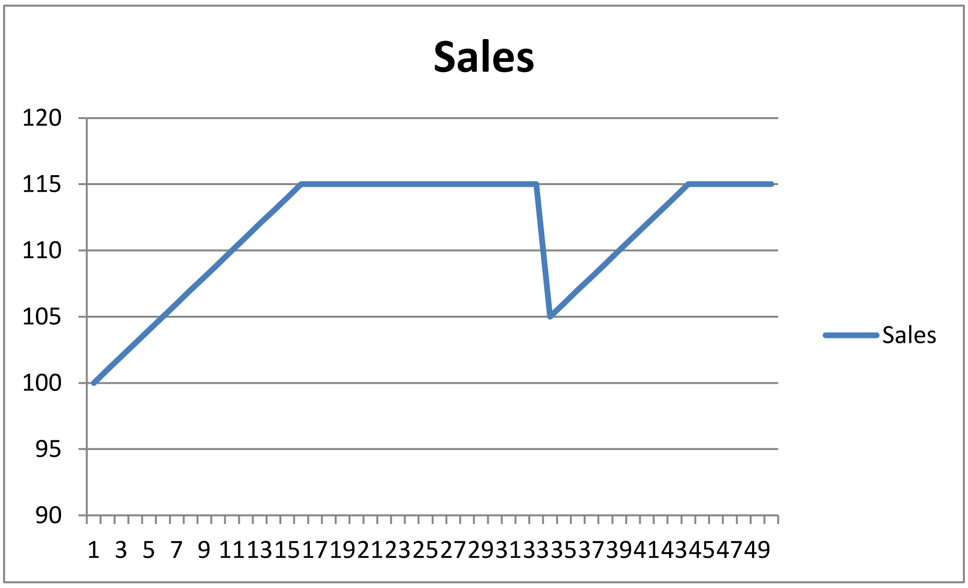
nSales=nCustomer-(nPrice-nOriginalPrice);

这个函数的意义在于，当价格上升时，销量会减少，且减少的部分恰好等于价格上升的部分。价格每上升 1 单位，就少 1 单位销量，反之亦然。在这个模型中，商家的价格只要变化都会引起销量的变化，整个市场对于价格信息的反馈是极为敏感而且理性的。

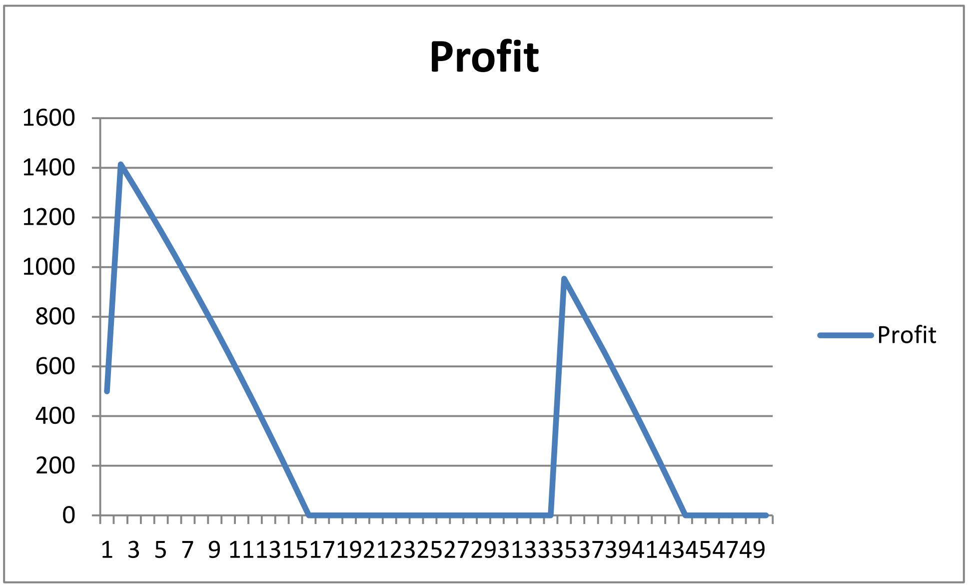
参数设置为：消费者：100，初始价格：25，初始质量：优质，初始销量：100，优品陈本：20，劣品成本为 10，价格变动幅度：1

程序运行结果如下：

价格图的趋势比较明了简单，价格在持续下降后形成了波动的周期，即在优质商品成本与劣质商品成本之间波动的周期。分析原因，可能为价格不断下降，但是检验器保证价格停在了劣质商品的成本。随后又是检验器，在商家随机生产出优质商品时将价格抬升到了优质商品的成本价。随后商家继续生产劣质商品，并重复这一价格变动周期。



很有趣的是，销量曲线几乎就是将价格曲线反过来的样子，这也体现了消费者对于价格的偏好，低价对应高销量，高价对应低销量。销量周期的出现原因应当与价格周期的出现原因相同。这一周期的出现并非是因为市场机制生成了商业周期，而是本模型的算法特点导致的，并非是经济学上的“周期”。



利润图中的波动情况与价格图中的波动情况完全一致。这周期是由于算法的特点而出现的，而非经济学原理。

质量图的结果印证了前面的所有猜测，价格上涨，利润上升，销量下降的原因都是因为 34 天出现了一次优质商品的生产。可以说，这个模型虽然体现出了消费者的价格偏好，但是并没有模拟出现实情况或者是经济学上的市场情况。该模型的仿真度是所有模型中最差的一个。

5.5.5　非完美信息有限理性模型

在前面尝试过 4 个模型后，结合自己外出购物时的心理活动体验，以及其他同学的感受，并融合了行为经济学的基础原理，我构造了第 5 个模型，也是仿真度最高的模型：非完美信息部分理性模型。

考虑到前面的结果，我需要让这一次的消费者反应函数符合信息不对称的

特征，只知道价格，不知道质量。还要具有有限理性，即需求随价格的变动不能太敏感。为此，我构造了如下的函数：

```
if (nPrice>nMaxPrice){nSales=nCustomer/2;}
else {nSales=nCustomer;}
if (nPrice<nMinPrice){nSales=nCustomer*2;}
```

这个函数的意思是说，消费者有一个价格接受的区间，有最大容忍价nMaxPrice,也有他能想出来的最低价位 nMinPrice,当高过最高价格时，销量会减少一半，当低过心理最低价位时，销量会成倍增加。这一函数利用了行为经济学中的一个概念：锚定效应。

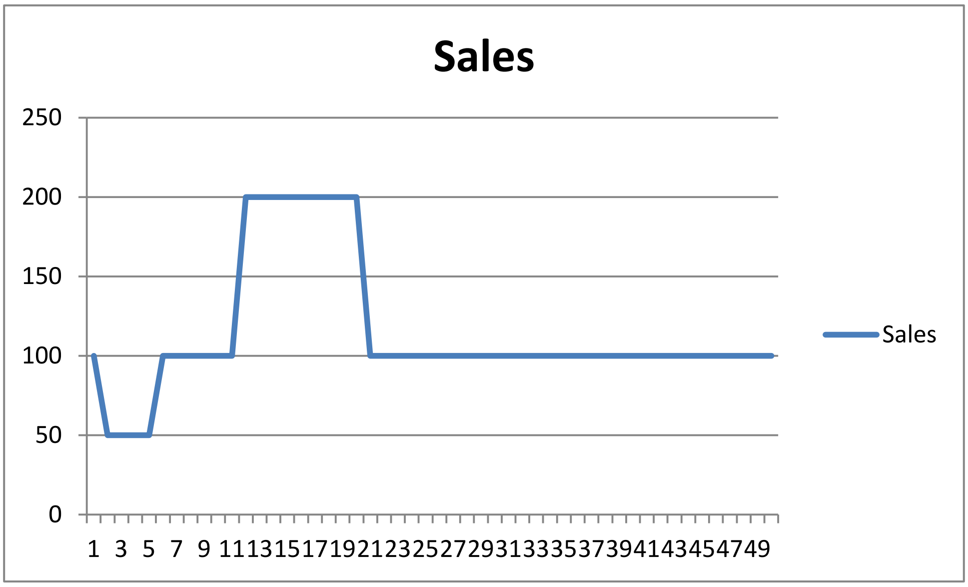
所谓锚定效应（Anchoring effect）是指当人们需要对某个事件做定量估测时，会将某些特定数值作为起始值，起始值像锚一样制约着估测值。在做决策的时候，会不自觉地给予最初获得的信息过多的重视。也就是说，初始的心理价位区间对需求量来说影响很大，而在这个区间内销量将保持平稳。这一行为特性符合我们在旅游购物中的心理状态。

参数设置：消费者：100，最大容忍价格：20，心理最低价格：17，初始价格：25，初始质量：优质，初始销量：100，优品成本：20，劣品成本：10，价格变动幅度：1

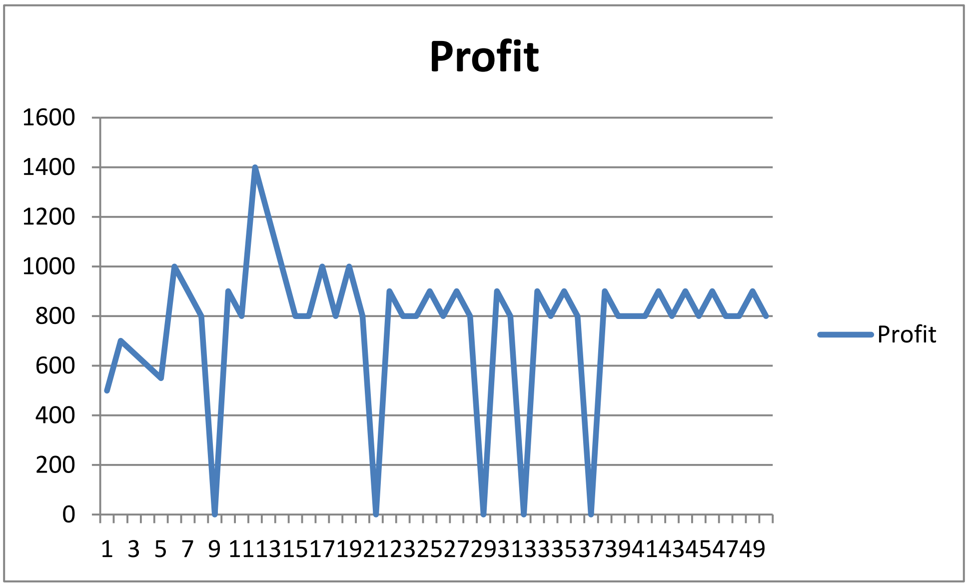
此模型结果如下图：

可以看出，价格首先有一段大的下降，随后稳定了一段时间，之后再上涨，并在（17,20）区间内波动。可以看到，价格波动的幅度在不断减小，周期也在

缩小。这说明，市场有形成均衡的趋势，且均衡价格略低于优品成本，远高于劣品成本。值得注意的是，这个模型形成的均衡价格是所有形成均衡的模型中最低的一个。从 40 天以后的折线可以看出，本模型最终的稳定程度可以和第 1 个模型媲美。

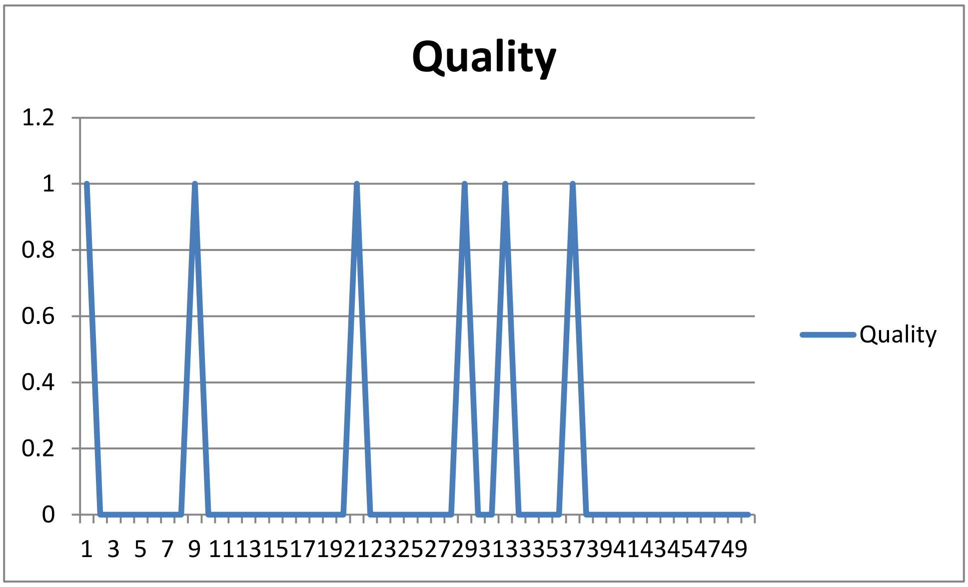


销量折线图突出来消费者反应函数的特点。当价格低于心理最低价位时，销量会成倍上升。相反，价格高于最大容忍价格时，销量便会减少 50%。所有这些减少的比例在模型中都是可以调整的。可以看出，与价格模型相对比，初始的高价位是导致销量下降的原因。随后的一段低价位稳定对应了其中 200 销量的稳定。随后价格再次上升，且逐步走向稳定。可以看到，在对应的天数中，销量也稳定在了一个正常的范围内。市场的均衡出现。



利润曲线的波动是比较大的。其中有 5 次零利润的情况出现，初步猜测是由

于商家提供了优质商品导致的。其中利润最大的 13 天的出现是因为高销量和高价格的出现。随后虽然有若干次利润为零的情况出现，但是从 40 天后没有出现此现象可以推断出，利润为零的情况在逐渐消失。商家在不断优化自己的策略。



质量图验证了利润图得出的结论，所有提供优质商品的天数都对应了零利润的出现。综合来看，该模型提供优质商品的趋势在不断减少，商家为了谋求更高的利润和销量倾向于提供劣质的商品。我认为该模型比较好的反应了旅游纪念品市场中低价劣品泛滥现象的演化过程。通过对这几个参量的比较，我可以运用经济学的知识系统的分析出这一现象出现的原因及解释。

有限理性模型，意味着消费者对于价格在某一区间内的变动并不敏感。然而如果在区间外，消费者的需求量是会呈几何级数变化的。这一特征导致商家在优化自己的策略时努力保证价格在这一区间内，以谋求稳定的客源。同时，为了实现在该区间内利润的最大化，价格应无限趋近于最大容忍价格，但不高于它，同时保证提供劣品，以降低成本。从上图来看，这两点都在此模型中得到了集中的体现。

第 6 章　　讨论

6.1　讨论

从静态模型的分析来看，商家提供低价劣品的动机是出于获得更高的利益。同样，消费者为了获得更高的消费者效用，也倾向于购买低价劣品，虽然他们并不知道商品的质量。

虽然这个模型可以部分的解释市场出现这种现象的原因是因为出现了纳什均衡，即任何一方单方面改变自己的策略都会减少自己的收益。然而，这个模型是一个一次博弈，真实的市场中交易是不断在进行的。所以静态博弈的结果并不能完全解释这一现象出现的原因。

类型	价格	利润	销量	质量
机理模型	波动较小，形成了均衡	初期有为零的情况，但趋于平稳	平稳，达到均衡	初期提供过优品，后期提供劣品
完全信息完全理性模型	价格波动较大，但形成了均衡	利润波动极大，呈周期性变化，多次出现零利润	初期平稳，后期出现零销量并有波动，总体平稳	多次出现优质商品，呈周期性出现，优品数量最多
质量偏好模型	价格初始波动大，后期形成波动小的均衡	后期利润稳定在零利润	同利润，后期销量为零	初期提供劣品，后期开始频繁提供优品
价格偏好模型	价格波动极大，未形成均衡	利润波动极大，未形成均衡	销量波动极大，未形成均衡	几乎全为劣品，只出现了两次优品
非完美信息有限理性模型	价格初期波动大，后形成稳定的价格均衡	初期波动大，有零利润出现，后逐步稳定	销量初期大，后期减少，出现均衡	劣品数量占主流，优品数量少，且趋势为提供劣品

上表为整理后的 5 种模型的结果。在构造了 5 种不同的动态模型后，我们可以从最后一个模型同其他模型的对比中看出：市场信息的不对称和消费者的有限理性是导致市场出现低价劣品均衡的主要原因。在市场信息完全对称且消费者完全理性的第 2,3,4 模型中，可以看到第 2 个模型同样出现了均衡，这可以说明传

统经济学理论中的均衡是存在的。同时在这一模型中出现了商业周期的现象。可以说，这一模型是很符合传统经济学理论的一个模型。并且，该模型提供了为数不少的优质商品。然而，这一模型最终的利润稳定在了零，并且其假设从根本上就决定了其本身就是不符合现实的。

在观察第 3,4 模型后，信息的完全对称和消费者的完全理性所带来的市场结果就更为可笑了。在市场完全偏好质量时，市场的确出现了所有模型中数量最多的优质商品，然而商家的利润却持续为零。很明显，这一现象在现实中根本不可能出现。商家一定会立即退出市场。而在偏好价格的第 4 模型中，价格呈现周期性的波动，且极有规律。然而利润的波动实在是太大了，而且有长段的为零情况。所以这一模型也是不符合现实的。

第 5 个模型关于信息不对称的假设以及消费者有限理性的设定都在最终的模型结果中得到了体现。整体价格水平最终呈现出了很好的均衡，利润为零的情况在减少，同时提供优质商品的周期在延长，销量也保持了相当的稳定。可以说，这一模型很好的仿真了旅游纪念品市场的交易状况。

由上可以得出结论，信息的不对称性和消费者的有限理性的确是导致旅游纪念品市场出现低价劣品泛滥这一现象的内在原因。在这一市场中，买家利用对产品信息了解的优势，消费者的有限理性带来的可利用的漏洞，以及商家本身对于更高利益的追求，共同导致了这一现象的出现。也可以说，这种现象是市场机制调配资源的结果。

同时，其他模型的结果还表明，在消费者完全理性的假设下，当市场中存在单一偏好的理性消费者且市场信息完全时，市场机制是无法发挥其合理调配资源的能力的。只有当市场中消费者偏好不同，且比例相对稳定时，市场中才可能出现均衡的情况。并且，当消费者出现有限理性的行为时，市场中同样可以出现均衡。虽然消费者完全理性且信息完全对称时出现的优品数量最多，消费者和商家的获益都最大，可正是由于该市场中存在的信息不对称以及有限理性，导致双方无法达成这一均衡。相对的，市场中存在信息不对称及有限理性时，同样可以形成均衡，只不过市场的趋势是提供劣质商品，而这与现实中的现象一致。可以说，此次模型比较好的仿真了真实市场的状态。

这些结论对于人们进一步探求市场机制是如何做到合理配置资源的有着很

重要的参考价值。毕竟，市场机制是目前人们找到的配置资源最合理，有效的方式。学习其内部运行原理对于人们更好的利用这一机制来避免资源的闲置与浪费有着重要的意义。

6.2　研究创新

本次研究成功的将信息不对称和消费者的有限理性用算法的形式表达在一个数学模型中。同时，本文给出了旅游纪念品市场中低价劣品泛滥的内在原因，并且通过对一些已有的经济学假设进行建模分析，得出了一些市场机制方面的结论，对于人们进一步研究市场经济的机制有着一定的指导和参考价值。同时，作者运用博弈论的思想来设计模型，用行为经济学的概念和观点来解释现象。与此同时，本文还将以往主要用心理学来解释现象的行为经济学通过数学建模的方式进行研究，这就把该学科的研究范围从心理学扩展到了数学的领域。

第 7 章　　后记

此次研究只进行了半年的时间，虽然只是一个市场的仿真模型，但可以说我投入了很多的精力和时间。我为自己能够通过研究提高自己的能力而感到骄傲，更为自己的研究有一定的意义感到自豪。

在研究的过程中，刘来福教授在研究的方法及思路上给了我很多的帮助和启发。他总能用风趣幽默的语言让我理解很深刻的概念和知识。我的论文也得到了他认真的审阅和批改。刘教授严谨的学术精神和诲人不倦的品格足以令我终身铭记。

由于此次研究涉及到程序的编写，我需要一位能指导我编程的老师。我校的赵雪春老师就一直在热心帮助我的编程工作。赵老师在外地，每晚都会和我讨论算法的编写，改进问题。最终程序得以正常运行，也得益于赵老师的帮助。感谢赵老师对我悉心的指导。

在研究的开题和论文的写作过程中，我都得到了王长沛教授的指导。王教授的思路开阔，交流直接，表达简洁明了。并在最后论文的写作中提出了很多很有价值的建议。谢谢您的建议。

感谢所有指导过我的老师，您们不仅让我获得了学业上的长进，更为我树立了治学为人的精神榜样。

最后，还要感谢我的父母。没有您们的支持和帮助，很难想象这一研究项目能够顺利完成。

写作论文是一个学习知识，交流看法的过程。由于研究时间仓促，以及本人水平、学识有限，其中谬误难免，材料也肯定有片面之处，恳请各位多多指教。

北京五中 高二七班 郭宇锋

2013 年 8 月于北京

附录 A　原始模型 Java 代码

```
import java.io.*;
public class 原始机理模型 //文件名
{
public static void main(String args[]) throws IOException //程序执行起点
 {
 int i;
 int nCustomer;//消费者数量
 int nCostPerUnit;//单位商品成本
 int nCostA; //优质商品成本
 int nCostB; //劣质商品成本
 int nDays;//运行天数
 int nSales;//销售量
 int nPreviousSales; //上一轮销售量
 int nPreviousQuality; //上一轮质量
 int nQuality; //商品质量
 int nPriceChanges;//价格变动幅度
 int nProfit; //利润
 int nPrice; //价格
 int nPreviousPrice; //上一轮价格
 int nPreviousProfit; //上一轮利润
 int nProfitIncreased;//判断利润是否增加的变量
 int nCurrentState;//市场状态变量，由特征值变化而来，是二进制数转换为十进制的数
 int nDecision; //决策变量，由两位数组成，第 1 位记录价格，第 2 位记录质量，0 为涨价，1 为不变，2 为降价，1 为优品，0 为劣品
 double nDecreasingVar;//递减系数
 double nSuitable; //适应度指数，为考察分类器适应情况设置，函数为nSuitable=nDecreasingVar*nSuitable+nProfit,nDecreasingVar 越小，则本期适应度受上期适应度值的影响就越小
 double probability; //概率
 InputStreamReader reader=new InputStreamReader(System.in);
 BufferedReader input=new BufferedReader(reader);
  System.out.println("请输入消费者数量=");
  String s1=input.readLine();
 nCustomer=Integer.parseInt(s1); //设定初始消费者数量
 System.out.println("请输入初始价格=");
 String s3=input.readLine();
 nPrice=Integer.parseInt(s3); //设定初始价格
 System.out.println("请输入初始销量=");
 String s4=input.readLine();
 nSales=Integer.parseInt(s4); //设定初始销量
 System.out.println("请输入天数=");
 String s5=input.readLine();
 nDays=Integer.parseInt(s5); //设定销售天数
```

```
        System.out.println("请输入价格变动幅度=");
        String s6=input.readLine();
        nPriceChanges=Integer.parseInt(s6); //设定价格变动幅度
        System.out.println("请输入递减系数=");
        String s7=input.readLine();
        nDecreasingVar=Double.parseDouble(s7); //设定递减系数
        System.out.println("请输入适应度=");
        String s8=input.readLine();
        nSuitable=Double.parseDouble(s8); //设定适应度值
        System.out.println("请输入初始质量(1 为优品，0 为劣品)=");
        String s9=input.readLine();
        nQuality=Integer.parseInt(s9); //设定初始质量
        System.out.println("请输入优品成本=");
        String s10=input.readLine();
        nCostA=Integer.parseInt(s10); //设定优品成本
        System.out.println("请输入劣品成本=");
        String s11=input.readLine();
        nCostB=Integer.parseInt(s11); //设定劣品成本
        nPreviousSales=nSales;
        nPreviousQuality=nQuality;
        nPreviousPrice=nPrice;
        nCostPerUnit=0;
        nPreviousProfit=nSales*(nPrice-nCostPerUnit);
        nDecision=01; //设定初始参数值
        double
nPStates[][]={{1/3,1/3,1/3,1/2,1/2,nSuitable},{1/3,1/3,1/3,1/2,1/2,nSuitable},{1/3,1/3,1/
3,1/2,1/2,nSuitable},{1/3,1/3,1/3,1/2,1/2,nSuitable},{1/3,1/3,1/3,1/2,1/2,nSuitable},{1/3
,1/3,1/3,1/2,1/2,nSuitable},{1/3,1/3,1/3,1/2,1/2,nSuitable},{1/3,1/3,1/3,1/2,1/2,nSuitabl
e},{1/3,1/3,1/3,1/2,1/2,nSuitable},{1/3,1/3,1/3,1/2,1/2,nSuitable},{1/3,1/3,1/3,1/2,1/2,n
Suitable},{1/3,1/3,1/3,1/2,1/2,nSuitable},{1/3,1/3,1/3,1/2,1/2,nSuitable},{1/3,1/3,1/3,1/
2,1/2,nSuitable},{1/3,1/3,1/3,1/2,1/2,nSuitable},{1/3,1/3,1/3,1/2,1/2,nSuitable}}; //程序
的核心所在，所有分类器的概率向量所在的二维数组
        for (i=1;i<nDays+1;i++) //开始进行交易行为模拟
        {
          nCurrentState=0; //初始化市场状态
          if (nQuality==1)
                    {nCostPerUnit=nCostA;}
              else {nCostPerUnit=nCostB;} //根据商品质量设定商品成本
             nProfit=nSales*(nPrice-nCostPerUnit); //销售利润的计算
          if(nProfit>nPreviousProfit)
          {nProfitIncreased=1;}
          else {nProfitIncreased=-1;} //定义销售利润状态
        switch (nDecision){ //根据获利情况，选择对应的上轮抉择进行概率向量的修改
                case 01:
```

```
nPStates[nCurrentState][0]+=0.1*nProfitIncreased;
nPStates[nCurrentState][1]-=0.05*nProfitIncreased;
nPStates[nCurrentState][2]-=0.05*nProfitIncreased;
nPStates[nCurrentState][3]+=0.1*nProfitIncreased;
nPStates[nCurrentState][4]-=0.1*nProfitIncreased;
nPStates[nCurrentState][5]=nDecreasingVar*nPStates[nCurrentState][5]+nProfit;
break;
case 00:
nPStates[nCurrentState][0]+=0.1*nProfitIncreased;
nPStates[nCurrentState][1]-=0.05*nProfitIncreased;
nPStates[nCurrentState][2]-=0.05*nProfitIncreased;
nPStates[nCurrentState][3]-=0.1*nProfitIncreased;
nPStates[nCurrentState][4]+=0.1*nProfitIncreased;
nPStates[nCurrentState][5]=nDecreasingVar*nPStates[nCurrentState][5]+nProfit;
break;
case 11:
nPStates[nCurrentState][0]-=0.1*nProfitIncreased;
nPStates[nCurrentState][1]+=0.05*nProfitIncreased;
nPStates[nCurrentState][2]-=0.05*nProfitIncreased;
nPStates[nCurrentState][3]+=0.1*nProfitIncreased;
nPStates[nCurrentState][4]-=0.1*nProfitIncreased;
nPStates[nCurrentState][5]=nDecreasingVar*nPStates[nCurrentState][5]+nProfit;
break;
case 10:
nPStates[nCurrentState][0]-=0.1*nProfitIncreased;
nPStates[nCurrentState][1]+=0.05*nProfitIncreased;
nPStates[nCurrentState][2]-=0.05*nProfitIncreased;
nPStates[nCurrentState][3]-=0.1*nProfitIncreased;
nPStates[nCurrentState][4]+=0.1*nProfitIncreased;
nPStates[nCurrentState][5]=nDecreasingVar*nPStates[nCurrentState][5]+nProfit;
break;
case 21:
nPStates[nCurrentState][0]-=0.1*nProfitIncreased;
nPStates[nCurrentState][1]-=0.05*nProfitIncreased;
nPStates[nCurrentState][2]+=0.05*nProfitIncreased;
nPStates[nCurrentState][3]+=0.1*nProfitIncreased;
nPStates[nCurrentState][4]-=0.1*nProfitIncreased;
nPStates[nCurrentState][5]=nDecreasingVar*nPStates[nCurrentState][5]+nProfit;
break;
case 20:
nPStates[nCurrentState][0]-=0.1*nProfitIncreased;
nPStates[nCurrentState][1]-=0.05*nProfitIncreased;
nPStates[nCurrentState][2]+=0.05*nProfitIncreased;
nPStates[nCurrentState][3]-=0.1*nProfitIncreased;
```

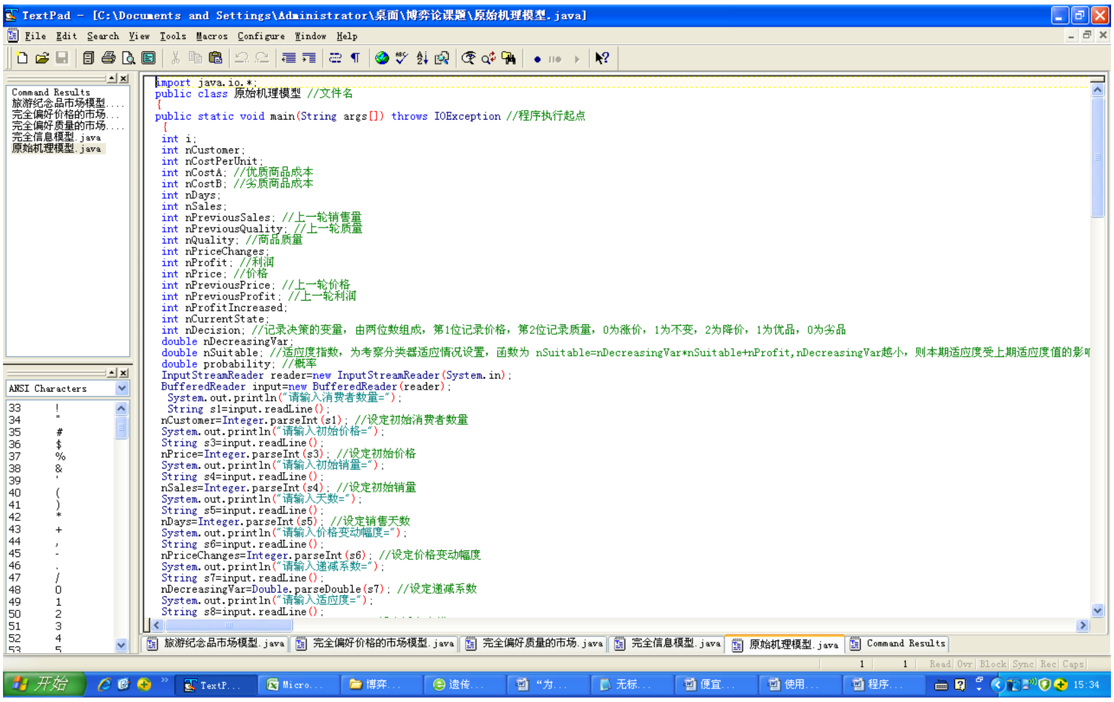
```
            nPStates[nCurrentState][4]+=0.1*nProfitIncreased;
        nPStates[nCurrentState][5]=nDecreasingVar*nPStates[nCurrentState][5]+nProfit;
            break;
            default:}

if(nPStates[nCurrentState][0]>0.9){nPStates[nCurrentState][1]+=(nPStates[nCurrentState
][0]-0.9)/2;nPStates[nCurrentState][2]+=(nPStates[nCurrentState][0]-0.9)/2;nPStates[nCu
rrentState][0]=0.9;}
if(nPStates[nCurrentState][0]<0.1){nPStates[nCurrentState][1]-=(0.1-nPStates[nCurrentSt
ate][0])/2;nPStates[nCurrentState][2]-=(0.1-nPStates[nCurrentState][0])/2;nPStates[nCur
rentState][0]=0.1;}
if(nPStates[nCurrentState][1]>0.9){nPStates[nCurrentState][0]+=(nPStates[nCurrentState
][1]-0.9)/2;nPStates[nCurrentState][2]+=(nPStates[nCurrentState][1]-0.9)/2;nPStates[nCu
rrentState][1]=0.9;}
if(nPStates[nCurrentState][1]<0.1){nPStates[nCurrentState][0]-=(0.1-nPStates[nCurrentSt
ate][1])/2;nPStates[nCurrentState][2]-=(0.1-nPStates[nCurrentState][1])/2;nPStates[nCur
rentState][1]=0.1;}
if(nPStates[nCurrentState][2]>0.9){nPStates[nCurrentState][1]+=(nPStates[nCurrentState
][2]-0.9)/2;nPStates[nCurrentState][0]+=(nPStates[nCurrentState][2]-0.9)/2;nPStates[nCu
rrentState][2]=0.9;}
if(nPStates[nCurrentState][2]<0.1){nPStates[nCurrentState][1]-=(0.1-nPStates[nCurrentSt
ate][2])/2;nPStates[nCurrentState][0]-=(0.1-nPStates[nCurrentState][2])/2;nPStates[nCur
rentState][2]=0.1;}   //检验器算法，控制概率向量不会出现负值或大于 1 的情况
        if (nPrice>nPreviousPrice) {nCurrentState+=8;}
        else {nCurrentState+=0;} //市场上价格上涨
        if (nSales>nPreviousSales) {nCurrentState+=4;}
        else {nCurrentState+=0;} //市场上销量上涨
        if (nProfitIncreased==1) {nCurrentState+=2;}
        else {nCurrentState+=0;} //市场上利润增加
        if (nQuality<nPreviousQuality) {nCurrentState+=1;}
        else {nCurrentState+=0;} //市场上质量降低
          nPreviousSales=nSales;
          nPreviousQuality=nQuality;
          nPreviousPrice=nPrice;
          nPreviousProfit=nProfit; //初始化变量
          System.out.println("第"+i+"天的销量为:"+nPreviousSales);
          System.out.println("第"+i+"天的质量为:"+nPreviousQuality);
          System.out.println("第"+i+"天的价格为:"+nPreviousPrice);
          System.out.println("第"+i+"天的利润为:"+nPreviousProfit); //输出每天市场
相关变量的状况
          System.out.println(" ");
          probability = Math.random(); //生成随机数,开始商家的决策，分类器算法的
实现
          if (probability<nPStates[nCurrentState][0]) //进行第 1 次判断，是否涨价，
```

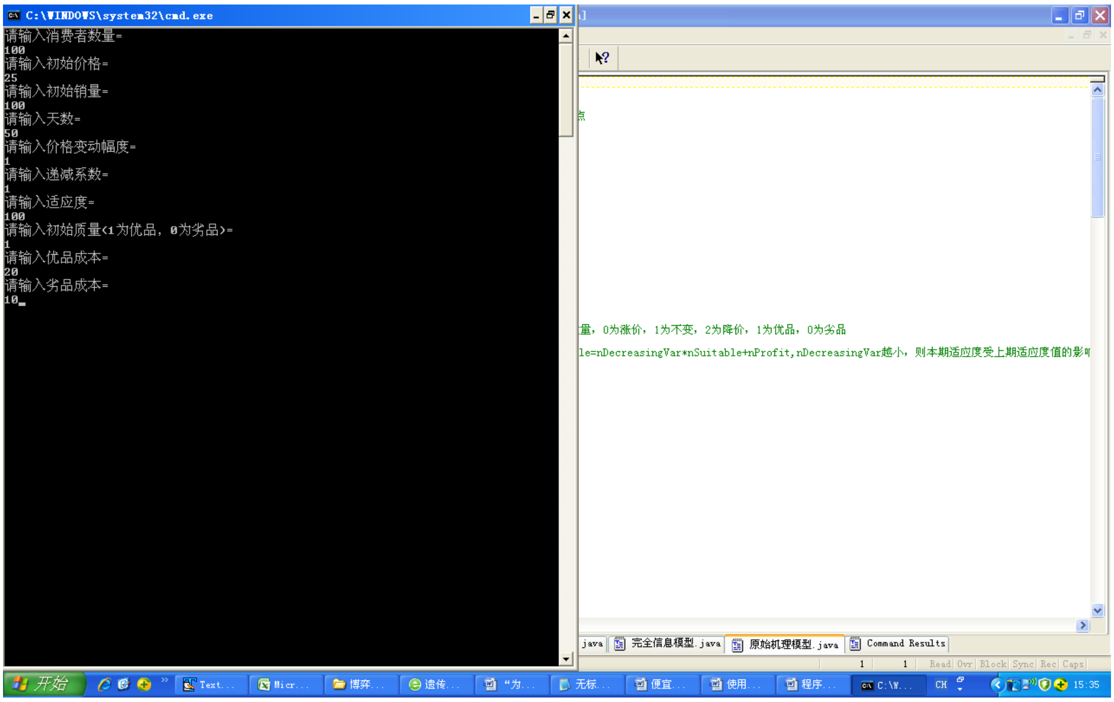
```
通过 2 维数组中存储的对应市场状态的概率向量进行决策
        { probability = Math.random(); //再次生成随机数
        if (probability<nPStates[nCurrentState][3]) //进行第 2 次判断，是否提供优
质商品
        {nDecision=01; //01 为涨价优品
        nPrice+=nPriceChanges; //涨价
        nQuality=1;} //提供优质商品
        else {nDecision=00; //判断不提供优质商品，00 为涨价劣品
        nPrice+=nPriceChanges;
        nQuality=0;}
        }
        else if (probability<nPStates[nCurrentState][0]+nPStates[nCurrentState][1])
        {probability = Math.random();
          if (probability<nPStates[nCurrentState][3])
          {nDecision=11; //价格不变，提供优质商品
          nQuality=1;}
          else {nDecision=10; //价格不变，提供劣质商品
          nQuality=0;}
        }
        else {probability = Math.random();
        if (probability<nPStates[nCurrentState][3])
        {nDecision=21; //降价，提供优质商品
        nPrice-=nPriceChanges;
        nQuality=1;}
        else {nDecision=20; //降价，提供劣质商品
        nPrice-=nPriceChanges;
        nQuality=0;}
        }
          if (nPrice<nCostPerUnit) {nPrice=nCostPerUnit;}
       nSales=nCustomer;
    }
    int m=0;
    int n=0;
    for(m=0;m<16;m++){
      for(n=0;n<6;n++)
    System.out.print(nPStates[m][n]+" "); //输出 2 维数组，观察各分类器概率向量情
况，并观察适应度高低
    }
    }
    }
```

附录 B　程序运行截图

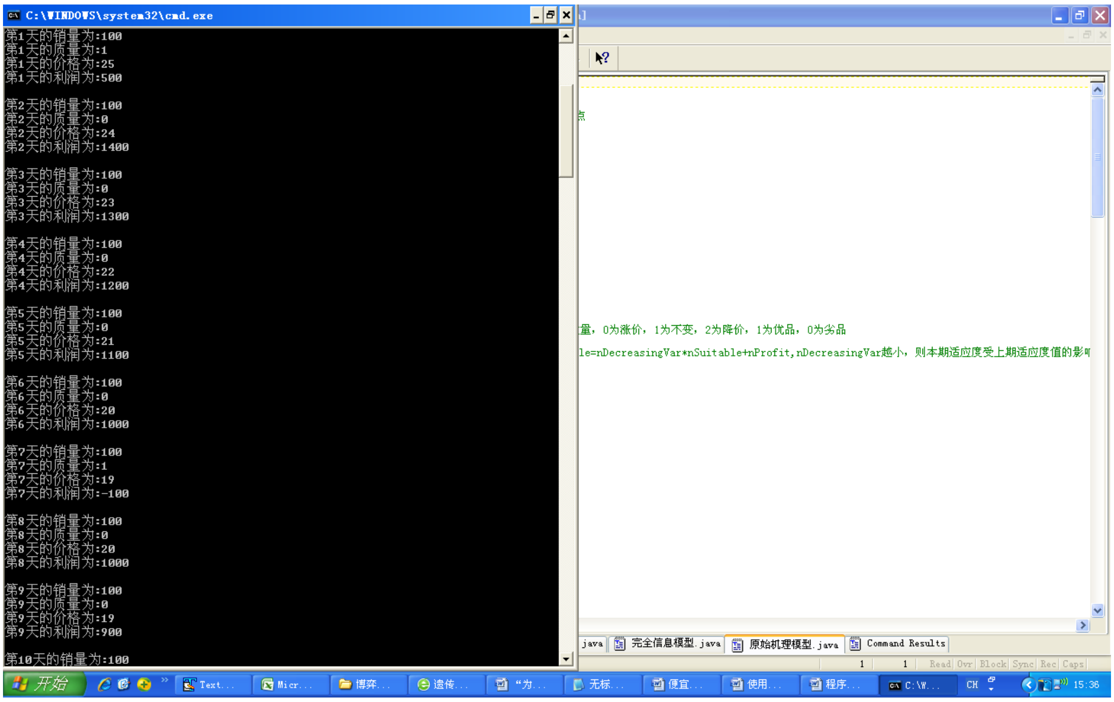
程序运行界面



程序编写界面



程序运行界面



结果输出界面

参考文献

1、《博弈实验进展》 葛新权，王国成编 社会科学文献出版社 2008-5-1

2、《36.5° 行为经济学》 李俊求（韩）著 史丽译 中信出版社 2010-12

3、《经济博弈论》 谢识宇编 复旦大学出版社 2002-1-1

4、《数理经济学教程》 纪玉山主编 科学出版社 2009-1

5、《基于消费者心理与行为的旅游纪念品营销策略研究》周静云，彭丽 湖南科技职业学院 《今日财富》2011 年 10 期 P34-P35

6、《高级人工智能（第三版）》 史忠植 科学出版社 2011-7-1